Details about the firmware extractor and decompressor by Colibri

Important note: All informations in this document are strictly for Educational purpose only. The author is not to be made liable for abuses. No warranty for the correctness of all data and procedures.

The bskyb digibox can get a new firmware via an over-the-air update via the Astra2 satellite. The firmware files are transmitted compressed over-the-air. The digibox will decompress the firmware file before it is stored on the flash. The following two chapters will show you how the extraction and decompression process works.

Chapter 1: How to extract firmware files from a logged stream

The firmware stream can be logged on Astra 2 (28.2 degrees east) from the BBC transponder 11720 H SR 27.5 MS/s FEC 2/3. The firmware PID is 60 hex (92 dec.). You can log the firmware stream e.g. with an DVBs card in you PC (I used this one http://www.hauppauge.com/html/dvb_s.htm). Additionally you need a loggingsoftware (I used "DVB Workshop" from http://www.dvingena.de.vu/).

The loggingsoftware produce a file with the following structur. The file contains many short blocks. Each block starts with a 00 byte, the next byte is the table_id with the value B5. The next two bytes (word) contain 4 bit RFU data and the 12 bit length of the following firmwaresection. Use “word AND 0x0FFF” to get the lengthinformation. Then the firmwaresection follows. Then fillbytes with the value FF are used before the next block starts. The count of the fillbytes are not specified, but you can recognize the end when the value is not FF but 00 B5 … (the startbytes of the next block).

Each firmware section consists of a 21 dec. byte header, an datablock and an 4 byte CRC.

The header starts with the sectiontyp of two bytes. If the sectiontyp is 0001 the datablock contains the real firmware file. If the sectiontyp is 0000 the datablock contains additional information to the firmware file. There are textstrings like “PACE” or “OS-BOOT” inside. If the sectiontyp is FFFF there are only the following three bytes in the datablock FF 1E F0.

After 4 unknown bytes the first byte of modelcode occurs. After one unknown byte the second byte of modelcode occurs. This two byte modelcode is also underneath each digibox. Search for a label that starts with “NDS:”. The first 4 chars after “NDS:” are the 2 byte modelcode. Example from an Silver BSKYB 2500N: “NDS:9F04xxxxxxxxxxxx”.

The following table shows which modelcode correspond to which model:

Panasonic
0F01: Black TU-DSB20
0F02: Silver TU-DSB30
0F03: Silver TU-DSB31 or DSB35

Pace
9F01: Black BSKYB 2200
9F02: Black BSKYB 2200
9F03: Black / Grey BSKYB 2500B
9F04: Silver BSKYB 2500N
9F05: Silver BSKYB 2500S3
9F06: Silver BSKYB 2500S4
9F20: Sky Plus box

Amstrad
4F01: Black DRX100
4F02: Black / Silver DRX100
4F03: Black / Silver DRX200

Sony
6F01: Silver 750
6F02: Silver 760

Grundig
4E01: Black GDS100
4E02: Black GDS100 / 200
4E03: Black or Silver GDS200
4E04: Silver GDS 310/2

After three unknown bytes the firmwareversion occurs. Based on this byte the digibox can decide if a new fireware is available.

The next DWORD (4 bytes) shows the fileoffset to which the filedata must be copyed.

The next DWORD (4 bytes) shows the filelength. With this information the digibox can decide if it have already received all filedata blocks.

Here is an example of an logged firmware stream from 2002-10-10. It contains three firmware files for PACE modelcode 9F01 (firmwareversion 8C), 9F02 (firmwareversion 66) and 9F03 (firmwareversion 3A). The 4 byte CRC at the end of each block is not shown.

Colors: Length, sectiontyp, modelcode, firmwareversion, fileoffset, filelength, Text

```
00 B5 8F FD 00 01 01 00 00 00 9F 00 02 00 00 00 66 00 0D F7 64 00 0E 26 AD <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 02 00 00 00 66 00 0E 07 48 00 0E 26 AD <filedata>
00 B5 8F 9A 00 01 01 00 00 00 9F 00 02 00 00 00 66 00 0E 17 2C 00 0E 26 AD <filedata>


00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 01 00 00 00 8C 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 01
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 13 54 00 00 00 01 00 2C 00 01 00 0E 07 D6 00 19 13 54 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 29 4B 19
B7 11 3F 59 6B 69 9D 48 6F 52 1B 99 1F B3 30 85 D4 5B 12 4D 04 9C 49 C6 F2 31 22 F9 DE 2F 41 A8 94
6D CB 13 A5 0E A3 5D 8B 52 1E 17 9F 2E F9 BB 6E E0 AF 6D B0 53 0E A5 F6 7B DE 8B 9B 2A DF 01 EB FE
6B 67 5B DF 5D 49 F4 6E 04 BD 94 62 58 A9 7D D9 11 D2 C9 14 D0 8A 60 8D 75 4A 19 8F DD 00 FF CB 2E
0F E3 39 64 F1 B0 65 AA DE 8A EC A2 9C 3E 58 C2 9D 98 16 23 4C A6 1D 98 9E 31 08 EB EF B0 15 7C 52
96 78 08 BA 00 C8 A2

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0
```

```
00 B5 81 0F 00 00 01 00 00 00 9F 00 02 00 00 00 66 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 02
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 46 BC 00 00 00 01 00 2C 00 01 00 0E 26 AB 00 19 46 BC 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 73 C9 27
27 5D 93 4A 37 3C A3 DC 2D 9D 1A AE BD AD BD 8E 05 F0 22 B7 55 D6 59 52 2A AD 7F 48 B5 43 BB BF 6F
26 0C B3 22 60 9C 5C 51 A7 06 87 E0 08 FB C6 FA 6C C4 9A B6 6F 04 15 86 78 33 38 3A E7 FD 33 99 5C
2E 4B 6C 80 6B E1 82 AE 59 9A FC 49 FA 11 37 6D B7 76 FB C1 00 28 06 4A FA F9 31 57 04 C3 62 70 5C
15 36 71 4E A0 C3 24 C4 FE 68 EE 0D 5B 4A 6A 2E 86 BA 3F 90 1B B3 03 AC 85 32 7B BD D1 37 7A 6E E9
0B 00 0C 31 00 A8 52

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 03 00 00 00 3A 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 03
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 69 98 00 00 00 01 00 2C 00 01 00 0E 3E 11 00 19 69 98 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 1A 87 A5
E9 B8 D1 30 FA E8 36 41 46 19 49 04 85 F3 43 5B 50 3E A8 25 3D 9C 99 15 05 0E 57 D3 CB BB 09 C4 F8
B5 E6 76 53 35 75 BE C8 D9 EF 35 DB D5 08 AF FA BB EF FF B4 A0 CA E8 2E 50 1A CC 92 09 D2 D2 D2 81
4C 5B 98 32 C6 6B A5 71 A3 EC B2 A2 AC 54 89 13 96 C9 D0 C5 CE 72 65 9E 0F 53 D0 8B 87 E0 63 BA C4
E2 DD 1D BA 72 65 8C 28 79 5F 5A D5 73 06 11 55 A1 4F 92 1A 7F EF 30 61 4A 31 75 52 09 E1 0A 59 EA
FF 2A 5C 94 00 0B 49

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0



00 B5 8F FD 00 01 01 00 00 00 9F 00 03 00 00 00 3A 00 00 00 00 00 0E 3E 13 <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 03 00 00 00 3A 00 00 0F E4 00 0E 3E 13 <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 03 00 00 00 3A 00 00 1F C8 00 0E 3E 13 <filedata>
…
00 B5 8F FD 00 01 01 00 00 00 9F 00 03 00 00 00 3A 00 0E 17 2C 00 0E 3E 13 <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 03 00 00 00 3A 00 0E 27 10 00 0E 3E 13 <filedata>
00 B5 87 38 00 01 01 00 00 00 9F 00 03 00 00 00 3A 00 0E 36 F4 00 0E 3E 13 <filedata>



00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 01 00 00 00 8C 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 01
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 13 54 00 00 00 01 00 2C 00 01 00 0E 07 D6 00 19 13 54 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 29 4B 19
B7 11 3F 59 6B 69 9D 48 6F 52 1B 99 1F B3 30 85 D4 5B 12 4D 04 9C 49 C6 F2 31 22 F9 DE 2F 41 A8 94
6D CB 13 A5 0E A3 5D 8B 52 1E 17 9F 2E F9 BB 6E E0 AF 6D B0 53 0E A5 F6 7B DE 8B 9B 2A DF 01 EB FE
6B 67 5B DF 5D 49 F4 6E 04 BD 94 62 58 A9 7D D9 11 D2 C9 14 D0 8A 60 8D 75 4A 19 8F DD 00 FF CB 2E
0F E3 39 64 F1 B0 65 AA DE 8A EC A2 9C 3E 58 C2 9D 98 16 23 4C A6 1D 98 9E 31 08 EB EF B0 15 7C 52
96 78 08 BA 00 C8 A2

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 02 00 00 00 66 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 02
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 46 BC 00 00 00 01 00 2C 00 01 00 0E 26 AB 00 19 46 BC 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 73 C9 27
27 5D 93 4A 37 3C A3 DC 2D 9D 1A AE BD AD BD 8E 05 F0 22 B7 55 D6 59 52 2A AD 7F 48 B5 43 BB BF 6F
26 0C B3 22 60 9C 5C 51 A7 06 87 E0 08 FB C6 FA 6C C4 9A B6 6F 04 15 86 78 33 38 3A E7 FD 33 99 5C
2E 4B 6C 80 6B E1 82 AE 59 9A FC 49 FA 11 37 6D B7 76 FB C1 00 28 06 4A FA F9 31 57 04 C3 62 70 5C
15 36 71 4E A0 C3 24 C4 FE 68 EE 0D 5B 4A 6A 2E 86 BA 3F 90 1B B3 03 AC 85 32 7B BD D1 37 7A 6E E9
0B 00 0C 31 00 A8 52

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 03 00 00 00 3A 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 03
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 69 98 00 00 00 01 00 2C 00 01 00 0E 3E 11 00 19 69 98 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 1A 87 A5
E9 B8 D1 30 FA E8 36 41 46 19 49 04 85 F3 43 5B 50 3E A8 25 3D 9C 99 15 05 0E 57 D3 CB BB 09 C4 F8
B5 E6 76 53 35 75 BE C8 D9 EF 35 DB D5 08 AF FA BB EF FF B4 A0 CA E8 2E 50 1A CC 92 09 D2 D2 D2 81
4C 5B 98 32 C6 6B A5 71 A3 EC B2 A2 AC 54 89 13 96 C9 D0 C5 CE 72 65 9E 0F 53 D0 8B 87 E0 63 BA C4
E2 DD 1D BA 72 65 8C 28 79 5F 5A D5 73 06 11 55 A1 4F 92 1A 7F EF 30 61 4A 31 75 52 09 E1 0A 59 EA
FF 2A 5C 94 00 0B 49

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0



00 B5 8F FD 00 01 01 00 00 00 9F 00 01 00 00 00 8C 00 00 00 00 00 0E 07 D8 <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 01 00 00 00 8C 00 00 0F E4 00 0E 07 D8 <filedata>
```

```
00 B5 8F FD 00 01 01 00 00 00 9F 00 01 00 00 00 8C 00 00 1F C8 00 0E 07 D8 <filedata>
…
00 B5 8F FD 00 01 01 00 00 00 9F 00 01 00 00 00 8C 00 0D E7 80 00 0E 07 D8 <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 01 00 00 00 8C 00 0D F7 64 00 0E 07 D8 <filedata>
00 B5 80 A9 00 01 01 00 00 00 9F 00 01 00 00 00 8C 00 0E 07 48 00 0E 07 D8 <filedata>



00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 01 00 00 00 8C 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 01
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 13 54 00 00 00 01 00 2C 00 01 00 0E 07 D6 00 19 13 54 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 29 4B 19
B7 11 3F 59 6B 69 9D 48 6F 52 1B 99 1F B3 30 85 D4 5B 12 4D 04 9C 49 C6 F2 31 22 F9 DE 2F 41 A8 94
6D CB 13 A5 0E A3 5D 8B 52 1E 17 9F 2E F9 BB 6E E0 AF 6D B0 53 0E A5 F6 7B DE 8B 9B 2A DF 01 EB FE
6B 67 5B DF 5D 49 F4 6E 04 BD 94 62 58 A9 7D D9 11 D2 C9 14 D0 8A 60 8D 75 4A 19 8F DD 00 FF CB 2E
0F E3 39 64 F1 B0 65 AA DE 8A EC A2 9C 3E 58 C2 9D 98 16 23 4C A6 1D 98 9E 31 08 EB EF B0 15 7C 52
96 78 08 BA 00 C8 A2

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 02 00 00 00 66 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 02
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 46 BC 00 00 00 01 00 2C 00 01 00 0E 26 AB 00 19 46 BC 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 73 C9 27
27 5D 93 4A 37 3C A3 DC 2D 9D 1A AE BD AD BD 8E 05 F0 22 B7 55 D6 59 52 2A AD 7F 48 B5 43 BB BF 6F
26 0C B3 22 60 9C 5C 51 A7 06 87 E0 08 FB C6 FA 6C C4 9A B6 6F 04 15 86 78 33 38 3A E7 FD 33 99 5C
2E 4B 6C 80 6B E1 82 AE 59 9A FC 49 FA 11 37 6D B7 76 FB C1 00 28 06 4A FA F9 31 57 04 C3 62 70 5C
15 36 71 4E A0 C3 24 C4 FE 68 EE 0D 5B 4A 6A 2E 86 BA 3F 90 1B B3 03 AC 85 32 7B BD D1 37 7A 6E E9
0B 00 0C 31 00 A8 52

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 03 00 00 00 3A 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 03
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 69 98 00 00 00 01 00 2C 00 01 00 0E 3E 11 00 19 69 98 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 1A 87 A5
E9 B8 D1 30 FA E8 36 41 46 19 49 04 85 F3 43 5B 50 3E A8 25 3D 9C 99 15 05 0E 57 D3 CB BB 09 C4 F8
B5 E6 76 53 35 75 BE C8 D9 EF 35 DB D5 08 AF FA BB EF FF B4 A0 CA E8 2E 50 1A CC 92 09 D2 D2 D2 81
4C 5B 98 32 C6 6B A5 71 A3 EC B2 A2 AC 54 89 13 96 C9 D0 C5 CE 72 65 9E 0F 53 D0 8B 87 E0 63 BA C4
E2 DD 1D BA 72 65 8C 28 79 5F 5A D5 73 06 11 55 A1 4F 92 1A 7F EF 30 61 4A 31 75 52 09 E1 0A 59 EA
FF 2A 5C 94 00 0B 49

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0



00 B5 8F FD 00 01 01 00 00 00 9F 00 02 00 00 00 66 00 00 00 00 00 0E 26 AD <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 02 00 00 00 66 00 00 0F E4 00 0E 26 AD <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 02 00 00 00 66 00 00 1F C8 00 0E 26 AD <filedata>
…
00 B5 8F FD 00 01 01 00 00 00 9F 00 02 00 00 00 66 00 0D F7 64 00 0E 26 AD <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 02 00 00 00 66 00 0E 07 48 00 0E 26 AD <filedata>
00 B5 8F 9A 00 01 01 00 00 00 9F 00 02 00 00 00 66 00 0E 17 2C 00 0E 26 AD <filedata>



00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 01 00 00 00 8C 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 01
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 13 54 00 00 00 01 00 2C 00 01 00 0E 07 D6 00 19 13 54 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 29 4B 19
B7 11 3F 59 6B 69 9D 48 6F 52 1B 99 1F B3 30 85 D4 5B 12 4D 04 9C 49 C6 F2 31 22 F9 DE 2F 41 A8 94
6D CB 13 A5 0E A3 5D 8B 52 1E 17 9F 2E F9 BB 6E E0 AF 6D B0 53 0E A5 F6 7B DE 8B 9B 2A DF 01 EB FE
6B 67 5B DF 5D 49 F4 6E 04 BD 94 62 58 A9 7D D9 11 D2 C9 14 D0 8A 60 8D 75 4A 19 8F DD 00 FF CB 2E
0F E3 39 64 F1 B0 65 AA DE 8A EC A2 9C 3E 58 C2 9D 98 16 23 4C A6 1D 98 9E 31 08 EB EF B0 15 7C 52
96 78 08 BA 00 C8 A2

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 02 00 00 00 66 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 02
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 46 BC 00 00 00 01 00 2C 00 01 00 0E 26 AB 00 19 46 BC 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 73 C9 27
27 5D 93 4A 37 3C A3 DC 2D 9D 1A AE BD AD BD 8E 05 F0 22 B7 55 D6 59 52 2A AD 7F 48 B5 43 BB BF 6F
26 0C B3 22 60 9C 5C 51 A7 06 87 E0 08 FB C6 FA 6C C4 9A B6 6F 04 15 86 78 33 38 3A E7 FD 33 99 5C
```

```
2E 4B 6C 80 6B E1 82 AE 59 9A FC 49 FA 11 37 6D B7 76 FB C1 00 28 06 4A FA F9 31 57 04 C3 62 70 5C
15 36 71 4E A0 C3 24 C4 FE 68 EE 0D 5B 4A 6A 2E 86 BA 3F 90 1B B3 03 AC 85 32 7B BD D1 37 7A 6E E9
0B 00 0C 31 00 A8 52

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0

00 B5 81 0F 00 00 01 00 00 00 9F 00 03 00 00 00 3A 00 00 00 00 00 00 00 F6 00 3C 00 34 00 00 00 03
FF FF FF FF FF FF FF FF FF FF FF FF 00 19 69 98 00 00 00 01 00 2C 00 01 00 0E 3E 11 00 19 69 98 00
00 00 30 4F 53 2D 42 4F 4F 54 00 42 4F 4F 54 00 00 00 00 A0 00 00 00 00 00 00 9F FF FF FF FF FF FF
FF FF FF FF FF FF 50 41 43 45 30 20 20 20 20 20 20 20 20 20 20 20 FF FF FF FF 01 11 01 0C 1A 87 A5
E9 B8 D1 30 FA E8 36 41 46 19 49 04 85 F3 43 5B 50 3E A8 25 3D 9C 99 15 05 0E 57 D3 CB BB 09 C4 F8
B5 E6 76 53 35 75 BE C8 D9 EF 35 DB D5 08 AF FA BB EF FF B4 A0 CA E8 2E 50 1A CC 92 09 D2 D2 D2 81
4C 5B 98 32 C6 6B A5 71 A3 EC B2 A2 AC 54 89 13 96 C9 D0 C5 CE 72 65 9E 0F 53 D0 8B 87 E0 63 BA C4
E2 DD 1D BA 72 65 8C 28 79 5F 5A D5 73 06 11 55 A1 4F 92 1A 7F EF 30 61 4A 31 75 52 09 E1 0A 59 EA
FF 2A 5C 94 00 0B 49

00 B5 80 1C FF FF 01 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 03 FF 1E F0



00 B5 8F FD 00 01 01 00 00 00 9F 00 03 00 00 00 3A 00 00 00 00 00 0E 3E 13 <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 03 00 00 00 3A 00 00 0F E4 00 0E 3E 13 <filedata>
00 B5 8F FD 00 01 01 00 00 00 9F 00 03 00 00 00 3A 00 00 1F C8 00 0E 3E 13 <filedata>
```

Chapter 2: How to decompress firmware files

Each compressed file does contain a 17 dec. byte header:

- The first word (4 bytes) contain the text “COMP”, that indicates that the sourcefile is compressed.
- The second word contain the sourcefilelength – 2 (the last two bytes of the sourcefile seems to be a chksum).
- The third word contain the sum of the length of the decompressed data and the length of the additional data that is just copied 1:1 from source to destination. This is also the destinationfilelength (the decompressed file).
- The fourth word contain the length of the decompressed data.
- The last headerbyte contain always the value 00.

After the header the compressed data follows. If the third header word (destinationfilelength) is greater than the fourth header word (decompresseddatalength) than the sourcefile also contains a second datablock a the end that is just appended to the destination. The length of this datablock is fourth header word – third header word.
If the length of the third header word and the length of the fourth header word are equal then the sourcefile only contain compressed data but no second datablock.

The decompressing process works as follows:
The controlbyteA is fetched from the source. This byte contains 8 bits that describes the next 8 steps of the decompression routine (1st step = MSB and 8th step = LSB). After the 8 bits are used up the next controlbyteA is fetched form the source. After each step the length of the decompressed data must compared to the fourth header word. If the are identical the decompression process is finished. Possibly a second datablock must appended to the destination now (as described above).

If the bit of controlbyteA is 1 then a byte is copied from the source to destination.
If the bit of controlbyteA is 0 then a range of bytes is duplicated in the destinationbuffer. The next sourcebyte(s) contain the offset from the current destinationbufferpos-1 and the

length of the range (copycount). A offsetvalue of 0 meens therefore the last byte written to the destinationbuffer will be duplicated (once if the copycount is 1 and twice if the copycount is 2 …).

An example the destinationbuffer contain "`12 34 56 78`" offset is 2 and copycount is 5.
After 1 copyed byte the destinationbuffer contains: `12 34 56 78 34`
After 2 copyed bytes the destinationbuffer contains: `12 34 56 78 34 56`
After 3 copyed bytes the destinationbuffer contains: `12 34 56 78 34 56 78`
After 4 copyed bytes the destinationbuffer contains: `12 34 56 78 34 56 78 34`
After 5 copyed bytes the destinationbuffer contains: `12 34 56 78 34 56 78 34 56`

The fetching of offset and copycount from the source works as follows:
To get the offset and copycount a multible of a nibble (4 bits) must loaded from the source. Therefore in addition to a sourcebufferpointer a nibbleflag must be used. If the nibbleflag is 1 than the sourcebufferpointer points to a byte whose hi-nibble is already used up.

If the bit of controlbyteA is 0 then the controlbyteB is fetched from source.
If nibbleflag is 0 then the controlbyteA is just loaded from the source.
If nibbleflag is 1 then the controlbyteA results from the lo-nibble of the current sourcepointer and the hi-nibble of sourcpointer + 1.
Example: Sourcebufferpointer points to "12 34 56 …" and nibble flag is 1. The controlbyteA contains therefore 32 (not 23! Because hi-nibble keeps the place).

If controlbyteA is 100xxxxx then copycount is 2 and offset =xxxxx (the lower 5 bits).

If controlbyteA is 1100xxxx then copycount is 3 and offset =xxxx (the lo-nibble).

If controlbyteA is 101xxxxx then one additionally nibble is fetched from source.
If the nibbleflag was 0 then the hi-nibble of the sourcebyte is fetched. Instead of incrementing the sourcepointer the nibbleflag is change to 1 now.
If the nibbleflag was 1 then the lo-nibble of the sourcebyte is fetched. The sourcebufferpointer is incremented and the nibblefalg is changed to 0.
copycount is 2 and offset =eeeexxxxx (eeee is the content of the extranibble and xxxxx are the lower 5 bits of controlbyteA).

If controlbyteA is 1101xxxx then one additionally nibble is fetched from source.
If the nibbleflag was 0 then the hi-nibble of the sourcebyte is fetched. Instead of incrementing the sourcepointer the nibbleflag is change to 1 now.
If the nibbleflag was 1 then the lo-nibble of the sourcebyte is fetched. The sourcebufferpointer is incremented and the nibblefalg is changed to 0.
copycount is 3 and offset =eeeexxxx (eeee is the content of the extranibble and xxxx is the lo-nibble of controlbyteA).

If controlbyteA is 1110xxxx then one additionally nibble is fetched from source.
If the nibbleflag was 0 then the hi-nibble of the sourcebyte is fetched. Instead of incrementing the sourcepointer the nibbleflag is change to 1 now.
If the nibbleflag was 1 then the lo-nibble of the sourcebyte is fetched. The sourcebufferpointer is incremented and the nibblefalg is changed to 0.

copycount is 4 and offset =eeeexxxx (eeee is the content of the extranibble and xxxx is the lo-nibble of controlbyteA).

If controlbyteA is 1111xxxx then one additionally nibble is fetched from source.
If the nibbleflag was 0 then the hi-nibble of the sourcebyte is fetched. Instead of incrementing the sourcepointer the nibbleflag is change to 1 now.
If the nibbleflag was 1 then the lo-nibble of the sourcebyte is fetched. The sourcebufferpointer is incremented and the nibblefalg is changed to 0.
copycount is 5 and offset =eeeexxxx (eeee is the content of the extranibble and xxxx is the lo-nibble of controlbyteA).

If controlbyteA is 0111xxxx then controlbyteB and controlbyteC are fetched from source.
If the nibbleflag was 0 then the controlbyteB and controlbyteC are normally fetched. If the nibbleflag was 1 then the lo-nibble of the sourcebyte is the lo-nibble of controlbyteB. The hi-nibble of the next sourcebyte is the hi-nibble of controlbyteB and the lo-nibble is the lo-nibble of controlbyteC the hi-nibble of the next sourcebyte is the hi-nibble of controlbyteC.
The controlbytes A-C have the following structur:
0111 xxxx xyyy yyyy yyyy yyyy
if the value of the 5 bits xxxxx is not 11111 then xxxxx + 3 = copycount and yyyyyyyyyyyyyyy is the offset.
But if the value of the 5 bits xxxxx is equal 11111 controlbyteD is fetched from the source.
If the nibbleflag was 0 then the controlbyteD is normally fetched. If the nibbleflag was 1 then the lo-nibble of the sourcebyte is the lo-nibble of controlbyteD. The hi-nibble of the next sourcebyte is the hi-nibble of controlbyteD.
The controlbytes A-D have the following structur:
0111 1111 1yyy yyyy yyyy yyyy zzzz zzzz
if the value of the 8 bits zzzzzzzz is not 11111111 then zzzzzzzz + 22 hex (34 dec.) = copycount and yyyyyyyyyyyyyyy is the offset.
But if the value of the 8 bits zzzzzzzz is 11111111 then controlbyteE and controlbyteF is fetched from the source.
If the nibbleflag was 0 then the controlbyteE and controlbyteF are normally fetched. If the nibbleflag was 1 then the lo-nibble of the sourcebyte is the lo-nibble of controlbyteE. The hi-nibble of the next sourcebyte is the hi-nibble of controlbyteE and the lo-nibble is the lo-nibble of controlbyteF the hi-nibble of the next sourcebyte is the hi-nibble of controlbyteF.
The controlbytes A-F have the following structur:
0111 1111 1yyy yyyy yyyy yyyy 1111 1111 ssss ssss ssss ssss
The value of sssssssssssssss + 121 hex (289 dec.) = copycount and yyyyyyyyyyyyyyy is the offset.

If controlbyteA is 0yyyxxxx then controlbyteB is fetched from source.
If the nibbleflag was 0 then the controlbyteB is normally fetched. If the nibbleflag was 1 then the lo-nibble of the sourcebyte is the lo-nibble of controlbyteB. The hi-nibble of the next sourcebyte is the hi-nibble of controlbyteB.
The controlbytes A-B have the following structur:
0yyy xxxx xxxx xxxx
The value of the 3 bits yyy + 3 = copycount and the 12 dec. bits is the offset.

Here is an example of an compressed file (20021010_PACE_9F01_8C.bin).

Colors: controlbyteA, controldataB, controldataC, copied source data

```
43 4F 4D 50 = “COMP”
00 0E 07 D6 = sourcefilelength - 2
00 19 13 54 = destinationfilelength
00 19 12 58 = decompresseddatalength
00 = must be 00
```

```
FF 20 20 22 08 4D 6F 6E 20 FF 53 65 70 20 33 30 20 30 FF 39 3A 30 36 3A 31 31 20
FF 32 30 30 32 0A 00 6D 61 F5 69 6E 70 62 83 2E 9D 72 FF 00 20 25 F8 40 D0 27 FE
FD 24 F2 FC 23 FF D1 85 70 7F 82 FA E0 10 4B 22 F1 40 F7 25 F4 27 2F F0 08 40 32
D7 71 A1 21 44 D1 AC 1A F0 30 71 F3 44 26 F2 2D B2 BE 22 80 4F EA FF 25 FE E0 C3
21 A2 70 DD 60 4C 87 24 40 F5 30 19 21 …
```

For corrections and additional infos that you could send to the following email address, many thanks in advance:
colibri_dvb@lycos.com

You can find the newest document at the following page (also the two tools “FW-Extractor.zip” and “OTA-FW-Decompression.zip”):
http://colibri.move.to/